GRUNDLAGEN DER KÜNSTLICHEN INTELLIGENZ
Programming Exercise 4: Probability
Josefine Gaßner, Hanzhi Chen, Yuyin Lang

Submission deadline: XXXX th February 2022, 23:59
(last updated 14th January 2022)

Programing Exercise 4: Probability

Problem 1: Hidden Markov Model

Learning Outcomes:

- You can perform smoothing and prediction on a Hidden Markov Model.
- You can perform the Viterbi algorithm and find the most likely state sequence of a Hidden Markov Model.

We all know that COVID-19 has been spreading for more than two years by now. In order to fight the pandemic, many restrictions have been implemented and are changing constantly. As a result, conducting our exam on-site is in danger. We want to find out if we again will have to resort to an online exam.

To do so, we want to use an HMM modeling the spread of the virus with a method adapted from a recent paper [1]. In the paper, the authors learn the parameters for an HMM based on multiple nations' COVID-19 data to infer the severity state on smaller regions. They also visualize the time evolution and propagation across regions. The HMM has hidden states, called severity levels. Severity is derived by combining a region's cases and deaths. Severity 1 is the lowest, 7 the highest. The state transition diagram of the learned HMM is provided in Figure 1.
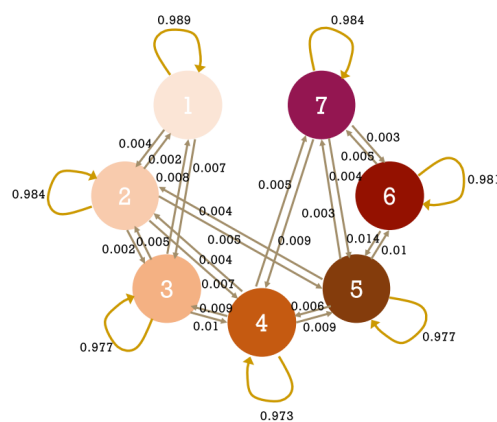

Figure 1: State transition diagram of the learned HMM.

For the observations, we collected the number of new cases and deaths for the regions of

Munich and Landkreis Munich[1] (where Garching is located).

A purely fictional Bavarian king, named Markus IV., rolled his seven-sided die and has it proclaimed that our exam can only take place on-site if the severity level in February 2022 is below 6 in Munich and Garching. Will he allow us to conduct the exam in person?

(Note: You do not have to read or understand the referenced paper in order to complete the exercise.)

*For more detailed information, see the provided jupyter notebook **HMM.ipynb**.*

Package Requirement

Make sure you install all the packages which are listed in requirement.txt in your jupyter notebook environment.

Passing criteria

Detailed information on the passing criteria of this programming exercise are given in the respective jupyter notebooks.

In short: the exercise is passed if the evaluation on Artemis of both problems is passed.

A pass will be awarded only if:

1. you submitted the **correct file** with the **correct name**.

2. you **did not change** the **variable names** provided by us within the template.

3. your submitted files can be run in an Anaconda environment (Python 3.7) with the packages provided by the *requirements.txt* **within a reasonable time (under 5 minutes)**.

4. like the rest of the programming exercises, this is an individual project and work **must** be your own. (We will use a plagiarism detection tool and any copied code will annul all bonus exercises from both the copier and the copied person!)

References

[1] Zhou, S., Braca, P., Marano, S., Willett, P., Millefiori, L. M., Gaglione, D., and Pattipati, K. R. (2021). Application of Hidden Markov Models to Analyze, Group and Visualize Spatio-Temporal COVID-19 Data. IEEE Access, 9, 134384-134401.

---

[1] https://github.com/jgehrcke/covid-19-germany-gae